Routing Of Leo-Based Satellite Using Guided Mode

M.Bhanusri, B.Alekhya, D.Anil Varma, Mr.Shaik Azeez

Abstract— Leo satellites are one of the most promising communication and plays important role in delivering high quality of data within the range of 500-1500 km. Leo satellites are for broadcasting internet and connecting remote places at the time of natural calamities. The Leo satellites are preferred due to less propagation delay and packet loss compared to Geo satellites. Generally the communication is done in indirect format means connecting the original satellite through intermediate ones. This leads to delay in process. To reduce that delay we are going to provide a direct communication to original satellite by knowing the co-ordinates of the particular satellites.

In order to find out the co-ordinates of the satellites we need the routing protocols. The routing protocols of the satellites are based on the quality of service, multicast and multipath. By using the algorithms like ATM and switching, satellite network topology we are going to provide a direct communication with less propagation delay. The multimedia applications and live transmissions cannot bear packet loss, jitter, and timeliness and need the high throughput through the transmission. We did a survey for the best routing protocol for satellite networks.

Index Terms— Routing, Satellite Networks, LEO, Quality of Service.

I. INTRODUCTION

Of late, low earth orbit (LEO) and medium earth orbit (MEO) based satellite networks have become the focus of attention as they promise lower delays and better bit-error rate performance for Internet and multi-media services than geostationary (GEO) satellites. LEOs and MEOs also provide higher bandwidth and high-speed links to end-users. Routing in such networks is difficult as their topology keeps changing dynamically (but in a predictable fashion). Several strategies have been proposed to deal with this

We feel that IP routing [10] for satellite networks is not efficient since it involves large overheads for each packet to be transmitted over the network. Thus, the bandwidth of the network may not be used efficiently. Also, IP switching (done over an ATM switch at the network layer) may not be efficient owing to the large number of routing tables that will have to be maintained on board each satellite.

ATM switching has similar disadvantages as IP switching. The solution to the routing problem given in [16,17] is based on ATM switching. But in no way does it discuss the feasibility of its implementation. Here, routing would be simple if a small number of switching tables needed to be maintained on board. Unfortunately, such is not the case. In a LEO network, the periodicity of each satellite is around 100 minutes, and after each revolution the satellite will be over a

M.Bhanusri, Student of LIET, Vizianagaram
B.Alekhya, Student of LIET, Vizianagaram
D.Anil Varma, Student of LIET, Vizianagaram
Mr.Shaik Azeez, Assistant professor, Vizianagaram

different point on earth (as the earth would have also rotated in the meantime). Also, there is a continuous slip between different satellite orbits. This means that it may take considerably longer than 24 hours before the same satellite configuration is repeated, i.e., the constellation periodicity is high. The constellation periodicity is defined as the time taken for the network configuration to get repeated taking into consideration the topology of the network, the position over the earth and time of day. In subsequent discussions, we shall refer to this as the periodicity of the LEO satellite network.Therefore, in the given scenario where we have close to 300 satellites in the network, the number of changes in the network topology that the constellation undergoes in a period is large. Consequently, the number of switching tables that will have to be maintained on board may be very large. This in turn increases the memory requirements of the satellites and does not seem feasible.A hybrid solution considering both ATM switching and IP switching has been given in [7], but this still suffers from the drawbacks that have been stated earlier.In this paper, we propose a routing strategy similar to ATM switching or simple tag switching (a virtual connection-oriented approach). However, we take into consideration the memory constraints of the satellites. We propose a stored program model that will not require all the necessary routing tables (or switching tables) on board the satellites all the time. In the process, we also provide a solution to the positioning problem of the Network Operations and Control Centers (NOCCs).

II. SYSTEM MODEL AND NETWORK TOPOLGY

A. System Model

Our system model is based upon the LEO satellite system proposed by Teledesic, which comprises a constellation of 288 satellites. The characteristic features of this constellation are listed below:

- Nominal altitude of 1375 km
- 12 orbital planes
- 24 satellites per plane
- Planes separated by 2 km in altitude
- Nominal inclination of 84.7 □
- Circular orbits
- Provides full earth coverage

Since the orbits vary in altitude, there is a continual slip between the orbital planes. So, as satellites pass between the poles, the inter-satellite link (ISL) between a pair of satellites may break and a new ISL may be formed with a different neighboring satellite.

B. Network Topology

The satellite constellation can be viewed as a network of 288

nodes with the ISLs representing the edges of the network. The topology of the network keeps on changing with time. The changes, however, are deterministic. The network goes through a long series of snapshots of the network topology.

A snapshot corresponds to the topology of the satellite network at a particular instant of time. When a new ISL is added or an already existing ISL is broken, a new snapshot is formed. Each snapshot has a finite lifetime with a given start and finish time.

The entire satellite constellation cycles through a series of snapshots. As stated earlier, the ISLs between satellites keep changing as they pass between the poles. Therefore, the number of snapshots generated in a cycle may be very large as the ISLs may keep changing pretty fast and the periodicity of the cycle is large. There may be just a few differences between successive snapshots. However, the shortest path between a pair of nodes in the network can be different between two successive snapshots. If the network is used to carry real-time traffic, then this may manifest itself as jitters at the time of snapshot transitions.

Our routing strategy is based on the concept of snapshots of the network. Note that for a given snapshot, each satellite may be over a different position over the earth, and at a different time of day. This will be discussed in the next couple of sections.

III. PROPOSED SATELLITE MODEL

The following aspects are considered while routing data at each node in the network:
•   the source of the incoming data

•   its destination

•   its type, and

•   its quantity

Here are the issues that lead us to formulate the satellite model:

1) As we consider the network configuration to be a series of snapshots changing over time, the satellite constellation is analogous to a fixed network during a particular snapshot. Therefore, routing boils down to having a suitable routing strategy for each snapshot and have a transition strategy from one snapshot to another. Hence, there would be a switching table for each snapshot.

2) The complexity of the routing issue lies in having high volumes of data with different QoS parameters, and the traffic density changing during the day.

3) Given the memory constraints on board the satellites and the large periodicity, it is not practical to have the switching tables for all the snapshots on board.

4) Since we can calculate all the snapshots off-line, we can have switching tables for just a few snapshots on board the satellites, and after a given period of time, we can flush out the old ones and upload switching tables for the next few snapshots required for routing.

5) The set of switching tables uploaded to a satellite from a ground station should be adequate to last until the satellite flies over the next ground station.

6) Thus, the satellites can be said to be analogous to stored-program computers and switching tables analogous to program instructions.

The stored-program model allows the satellites to have switching tables for just a few snapshots that are required for routing during a given time interval. We had mentioned in Section I that ATM switching is unfavorable owing to the large number of switching tables that will have to be stored on board for the entire lifetime of the satellites. The stored-program model, therefore, works out to be very efficient considering the fact that all the snapshots are not required all the time for routing, i.e., not all the tables need to be uploaded into the satellite memory. The details are given in Section IV (C).

The function of the nodes would be to just redirect the incoming data packets along the correct ISL by looking at the tag of the incoming packets and using the switching tables as mere look-up tables. Thus, the nodes are not involved in any computations and therefore, the memory on board can be largely used to store these tables.

IV. THE ALGORITHM

A. *Routing for a snapshot*

Data being carried by the satellite network can be classified into various QoS categories: hard real-time, soft real-time and non-real-time. Audio and video data are examples of hard real-time data that may be sent over the network. The data that applications like telnet and ftp send across the network can be termed as soft real-time. Electronic mail and news may be said to carry non-real-time data.

The routing problem can be viewed in two different ways – the shortest path problem or the multi-commodity flow problem.

Shortest path routing [2] between node pairs strives to minimize the latency by taking the minimum number of hops from the source to the destination. However, following this strategy to route all kinds of data may result in congestion in some parts of the network. This may result in low throughput of the network.

The problem can be alternatively thought of as a multi-commodity flow problem [2,6,9,11,14]. The aim of the multi-commodity flow problem is to maximize the network throughput. Latency is not a concern in this case. This approach cannot be adapted to route real-time data as the routing path may not necessarily be the shortest path. Thus, the latency can be substantially high. So, a suitable balance between the two routing policies has to be struck in order to achieve optimal results.

The balance can be achieved by dedicating a part of the bandwidth of each ISL for real-time data and all other data types sharing the remaining bandwidth. Thus, real-time data[1] can be routed using the shortest path technique while the other data types may be routed according to the multi-commodity flow approach.

The multi-commodity flow problem has the disadvantage of being computation intensive to obtain a solution. To solve this problem for a LEO satellite network with almost 300 nodes, obtaining an optimal solution for a single snapshot can take a considerably large amount of time. Moreover, if the switching tables have to be generated for all the snapshots using this approach, then the time complexity involved would be unpredictably high. Thus, even for the multi-commodity flow approach, approximations based on the shortest path solutions [3,5] that have lower time complexity will have to be considered. Therefore, the problem can be viewed as shortest path approximations.

The transition between snapshots, as mentioned before, may result in jitters. Therefore, it is not sufficient if we calculate only the shortest path between every pair of nodes for each snapshot. There should be alternative paths which when used, would result in smooth transition from one snapshot to another with as little jitters as possible. These alternative paths can also be used in case of node failure or congestion in the network. Therefore, the routing problem can be viewed as finding the k best shortest paths between every pair of nodes in the network.

B. Solution Approach

The problem now has been reduced to finding the k shortest paths between every pair of nodes in the satellite network. The strategy adopted to obtain the required switching tables is given below:

1) Solutions to the k shortest path problem [4,8,15, 18] are used to calculate k number of shortest paths for each source-destination pair for each snapshot, where $1 \leq k \leq n$ (n = number of satellites in the network).

2) The data stream is routed along the kth shortest path based on the latency and bandwidth requirements. Thus, real-time data may be routed along the shortest path and other kinds of data along the remaining (k-1) paths.

3) When a snapshot transition is nearing or when a transition has just occurred, instead of routing using the

4) k^{th} shortest path for that snapshot, a $k \square \delta$ shortest path is chosen. This is done to reduce the jitters that may be caused due to the sudden change in the shortest paths taken before and after the transition. The value of δ is

5) chosen such that the shortest path, which is to be selected in the next snapshot, produces as little jitter as possible. The alternate route may also be chosen in case of node failure or congestion at the node. For this, each satellite has to maintain the status of each of its ISL. The shortest path taken before and after the transition is shown in Figure 1. The paths are chosen such that the inter-arrival time (IAT) is reduced as far as possible, which in turn would reduce the jitters resulting from the transition.

---

[1] Finer granularity can be provided by further splitting the bandwidth between hard real-time and soft real-time traffic appropriately
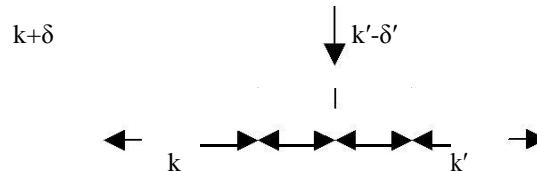
Transition Point

Figure 1

In the above figure, in order to reduce the jitter that may be caused due to the transition, alternate

shortest paths are selected for routing, such that $IAT(k) < IAT(k+\delta) < IAT(k'-\delta') < IAT(k')$. The

values of δ and δ' are chosen such that the transition from the kth shortest path to the k' shortest path is smooth.

4) Admission control has to be done at the nodes depending on the QoS parameter of the data to be uploaded. Each node has to maintain the status of all its ISLs and perform admission control appropriately. We shall, however, not go into its details.

5) The switching tables basically contain only the mappings for the incoming data at each node, which tell the next ISL to be taken towards the destination. Each table entry will contain the following fields: <ingress link, ingress label, egress link, egress label>. The table mapping is similar to tag switching [12] or multi-protocol label switching (MPLS) [1,13].

6) Each node in the network does the routing as follows:

• It looks at the tag of the incoming data packet.

• It then does a table look-up with the help of its ingress label (which indicates its QoS parameter).

• If the transition to the next snapshot is nearing or a transition has just occurred, then the shortest path is chosen as stated previously. If the ISL along which it is going to send the data is congested or the node at the other end has failed, then the next suitable shortest path is chosen.

• It then stamps the data packet with the egress label obtained from the table entry in place of its old ingress label.

• It then sends out the packet along the egress link obtained from the table entry.
This method of routing data is quite fast.

Identical routing decisions can be taken for all the packets of a stream. It also involves less computation overhead when compared to IP routing.

C. Ground Stations and Memory
Requirements

As mentioned earlier, only sets of switching tables will be

uploaded on to the satellites from time to time from ground stations. The NOCCs (Network Operations and Control Centers) that are going to be set up for billing and certain maintenance purposes can themselves act as these ground stations to store these switching tables. The set of tables which the satellites upload should be sufficient such that they can route using these tables till they pass over the next NOCC.

These NOCCs have to be located strategically for all the satellites to be able to upload the required switching tables from time to time. Every satellite in the network must have access to at least one NOCC to upload tables. The NOCCs will have to be positioned in such a way that a minimum number of them are required.

An important factor to be considered while estimating the number and positions of the NOCCs is fault tolerance. In the event of failure of an NOCC, a satellite must have the required tables for routing before it can pass over the next possible NOCC. Thus, the number of NOCCs required and their positioning assume a lot of importance for the stored-program approach.

The following factors are considered before deciding a suitable geographic location to position the NOCCs:

2)   The probability of each satellite passing over at least one of the NOCCs should be quite high. If the NOCCs are positioned near the equatorial region and only a small number of them are to be deployed, then the control centers will have to be widely spread out all along the equator reducing the possibility of a satellite passing over at least one of them. Therefore, a large number of control centers will be required so that at least one of them is visible to the satellites when they pass over the equator. This kind of placement is not that cost effective The locations for placement should be chosen such that there is low ingress (egress) traffic from (to) the environment and low data traffic to (from, respectively) the neighboring satellites. This is an important issue to be considered, because in regions where ground-satellite traffic is quite high, the bandwidth that would be used for uploading the tables from the NOCCs on to the satellites takes away from the available bandwidth for end-user traffic.

Considering the above factors, one such strategic position can be near both the poles. At the poles there will be low ingress (egress) traffic from (to, respectively) the environment. Also, the small surface area of the earth near the poles makes it possible to have an optimum number of NOCCs and also position them quite close to one another. This increases the probability of a satellite to view more than one control center to load its required tables in case one of them fails. As the periodicity of each satellite is around 100 minutes, it would be possible for each of the satellites to upload tables every 50 minutes if the NOCCs are deployed near both the poles.

In addition to the fact that less memory would be required on board the satellites if the switching tables were to be stored at ground stations, another big advantage is that the routing strategy can be easily modified by just switching to another set of switching tables. A new set of switching tables can be computed off-line. Then, at a predetermined time, these new tables could be uploaded instead of the tables generated for

the old strategy. This change in strategy is transparent to the satellites and does not require any hardware changes in them. The only job of the satellites is thus to upload sets of tables from the nearest NOCC, use them as look-up tables to route incoming data packets, flush them from memory after their finish time and upload a set of new switching tables. Thus, only enough memory on board the satellites is required to store these "look-up" tables.

If the approximate time for a satellite to pass over an NOCC is 50 minutes, and if the lifetime of a snapshot is around 5 minutes, then the total memory M (bytes) required on board the satellites can be calculated as follows:

M = (Table size) $\Box$ [(time between successive sets of snapshots ≈ 50)$\Box$ (snapshot lifetime ≈ 5)].

Now, each table entry consists of four integer values <ingress link, ingress label, egress link, egress label>. The number of table entries can be estimated to be around 50 -

6 ISLs $\Box$ 3 QoS parameters $\Box$ 3 shortest paths for each kind. Considering one word of storage for an integer,

Table size = (4 integers) $\Box$ (Number of entries ≈ 50)
= $4\Box\ 4\Box\ 50$

=   800 Bytes
∴ M = 800 $\Box$ 10 = 8 KB[2].

V.   CONCLUSION

We have given a virtual connection-oriented approach based on simple tag switching to route data in a LEO satellite network. We have removed the restriction of having all the switching tables on board, and have stated a feasible solution to the problem. Given all the snapshots on hand, and with all the switching tables generated for each snapshot, the concept of having just a few sets of switching tables at periodic intervals on-board the satellite looks highly feasible. This

The issue to be solved now will be to arrive at a suitable number of NOCCs to serve the satellite constellation. This issue needs to be addressed as the routing strategy is based on the assumption that each satellite has access to at least one NOCC to upload new tables as and when required. Also, simulations will have to be done in order to determine the exact number of snapshots, their lifetime and the table size for each of them.

This paper has thus been successful in chalking out a suitable routing technique taking into consideration the memory constraints of the satellites. It does this by mapping the routing table generation to the enumeration of the k shortest path for every snapshot. It also raises the issue of having the NOCCs being positioned at strategic positions and gives a probable answer.

---

[2] This is just an estimate. Even if the actual memory required is greater than the given estimate, it is still manageable provides the advantage of being able to change the routing policy in a transparent fashion.

REFERENCES

[1] R. Callon., P. Doolan, N. Feldman, A. Fredette, G. Swallow and A. Viswanathan, "A Framework for Multiprotocol Label Switching", Internet draft <draft-ietf-mpls-frmaework-00.txt>, May 1997.

[2] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, "An Introduction toAlgorithms", MIT Press and McGraw-Hill, 1990.

[3] D.A. Dunn, Wayne D.Grover and M.H. MacGregor, "Comparison of k Shortest Paths and Maximum Flow Routing for Network Facility Restoration", IEEE Journal on Selected Areas in Communication, Vol.12, pp.88-99, 1994.

[4] David Eppstein, "Finding the k shortest paths", Proc. 35th Symp. Foundations of Computer Science, IEEE, pp.154-165, 1994.

[5] S. Even, A. Itani and A. Shamir, "On the Complexity of Timetable and Multicommodty Flow Problem", S.I.A.M Journal on Computing, 5, (1976), pp.691-703.

[6] Lestor R. Ford, Jr., and D. R. Fulkerson, "Flows in Networks", Princeton University Press, 1962.

[7] Yukio Hashimoto and Behcet Sarikaya, "Design of IP-based Routing in a LEO Satellite Network", Proc. of the 3rd International Workshop on Satellite-Based Information Services, Mobicom'98, October 1998.

[8] Ernesto De Queiros Vieira Martins, Marta Margarida Braz Pascoal, and Jose Luis Esteves Dos Santos, "The K Shortest Paths Problem", International Journal of Foundations of Computer Science, June 1998.

[9] H. Nagamochi, T. Ibaraki, "On MFMC and Integral Flow Properties for Multicommodity Flow in Directed Networks", Lin. Alg. Appl., 114/115 (1989), 279-286.

[10] Paolo Narvaez, Antonio Clerget, and Walid Dabbous, "Internet Routing over LEO Satellite Constellations", Proc. of the 3rd International Workshop on Satellite-Based Information Services, Mobicom'98, October 1998.

[11] H. Okamura, "Multicommodity Flows in Graphs", Disc. Appl. Math. 6 (1983), 55-62.

[12] Y. Rekhter, B. Davie, D. Katz, E. Rosen and G. Swallow, "Cisco Systems' Tag Switching Architecture Overview", RFC 2105, Feb 1997.

[13] Eric C. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture", Internet Draft <draft-ietf-mpls-arch-02.txt>, July 1998.

[14] A. Schrijver, "Short Proofs on Multicommodity Flows and Cuts", 1989.

[15] Douglas R. Shier, "On Algorithms for Finding the k Shortest Paths in a Network", Networks Journal, Vol.9, No.3, 1979.

[16] Markus Werner, "A Dynamic Routing Concept for ATM-Based Satellite Personal Communication Networks", IEEE Journal on Selected Areas in Communications, Vol.15, No.8, October 1997.

[17] Markus Werner, Cecilia Delucchi, Hans-Jorg Vogel, Gerard Maral, and Jean-Jacques De Ridder, "ATM-Based Routing in LEO/MEO Satellite Networks with Intersattelite Links", IEEE Journal on Selected Areas in Communications, Vol.15, No.1, January 1997.

[18] J.Y. Yen, "Finding the k shortest loopless paths in a network", Management Science, 17:712-716, 1971